

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method in a data processing system having a program for allocating objects in a memory portion that includes a Young Generation and at least one Older Generation, said method comprising:

(a) for each of the objects, determining whether at least one object should ~~[[not]]~~ be allocated in said Young Generation in accordance with a first promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion;

(b) based on the determining whether each of the objects should be allocated using the first promotion policy, determining whether said at least one object should be allocated in said Young Generation in accordance with a second promotion policy ~~for said at least one object when said determining (a) determines that said at least one object should not be allocated in said Young Generation in accordance with said first promotion policy,~~ said second promotion policy exercised for at least postponing or cancelling promoting objects from said Young Generation to said Older Generation relative to the first promotion policy; and

(c) ~~storing said at least one object~~ each of the objects in said Young Generation in accordance with said first or second promotion policy ~~when said determining (b) determines the second promotion policy for said object,~~ and providing an indication of said second promotion policy in a header for a subset of the objects, the indication used during a promoting step to preempt use of the first promotion policy of said at least one object.

2. (Previously Presented) The method as recited in claim 2, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotion policy comprises:

determining whether said at least one object is used as temporary data.

3. (Previously Presented) The method as recited in claim 2, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotion policy comprises:

determining whether said at least one object is garbage.

4. (Previously Presented) The method as recited in claim 3, wherein said at least one object can be garbage within an acceptable time period.

5. (Previously Presented) The method as recited in claim 1, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotional policy comprises:

determining whether said at least one object is garbage.

6. (Previously Presented) The method as recited in claim 5, wherein said determining of whether said at least one object is garbage determines whether at least 50% of said at least one object is garbage.

7. (Previously Presented) The method as recited in claim 1, wherein said determining (a) of whether an object should be allocated in accordance with a first promotion policy comprises:

determining whether system code is allocating said object; and

wherein said method further comprises:

selecting a second promotion policy that postpones the promotion of said at least one object with respect to other objects allocated in accordance with said first promotion policy.

8. (Currently Amended) The method as recited in claim 1, wherein [[of]] said determining (a) of whether an object should be allocated in accordance with a first promotion policy comprises:

determining whether one or more of the following operations are being performed: loading a class, parsing a file that represents a class, dynamic compilation, and a call to a library function that generates temporary data.

9. (Previously Presented) The method as recited in claim 8, wherein said class is Java™ compliant class represented in a class file, said dynamic compilations are performed in Java™ compliant run time environment, and said library function is Java™ compliant library method.

10. (Previously Presented) The method as recited in claim 9, wherein said Java™ compliant library method is associated with concatenation of Java™ strings.

11. (Cancelled)

12. (Previously Presented) The method as recited in claim 1, wherein said header includes a preemption indicator that indicates a garbage collection count should be preempted and said at least one object should not be promoted to said next generation.

13. (Previously Presented) The method as recited in claim 1, wherein said header includes a preemption indicator and a preemption value;

wherein said preemption indicator indicates that a garbage collection count should be preempted;

wherein said preemption value provides a preemptive garbage collection count that is used instead of a garbage collection count.

14. (Previously Presented) The method as recited in claim 1, wherein said header provides a garbage collection count to determine when said at least one object should be promoted from said Young Generation to said Older Generation.

15. (Currently Amended) A computer system, comprising:
at least one processing unit;
a memory portion that is partitioned into a Young Generation and at least one Older Generation;
a first memory allocator that operates to allocate one or more objects in said ~~Older Generation from said~~ Young Generation in accordance with a first promotion policy for promoting said one or more objects from said Young Generation to an Older Generation in said memory; and
a second memory allocator that operates to allocate one or more other objects in said Young Generation in accordance with a second promotion policy for ~~cancelling or~~ at least postponing promoting said one or more other objects from said Young Generation to an Older Generation in said memory relative to the promoting performed in accordance with the first promotion policy, wherein the one or more other objects are being allocated by system code or are associated with an operation predefined as likely to generate garbage.
wherein,
~~in accordance with said second promotion policy, an indication of said second promotion policy is allocated in a header of said one or more objects.~~
16. (Currently Amended) The computer system as recited in claim 15, wherein said ~~method~~ system further comprises:
an allocation interface that can be used to access both said first and second memory allocators,
wherein,
said allocation interface operates to use said first or second memory allocators in accordance with an allocation selection, and
an allocation switching function that can be used to switch said allocation selection from said first memory allocator to said second memory allocator.
17. (Previously Presented) The computer system as recited in claim 15, wherein said first and second memory allocators can be directly accessed.

18. (Previously Presented) The computer system as recited in claim 15, wherein said processing unit:

determines whether at least one object should not be allocated using said first memory allocator in said Young Generation in accordance with said first promotion; and

determines said second promotion policy used by said second allocator when it is determined that at least one object should not be allocated in said Young Generation in accordance with said first promotion.

19. (Previously Presented) The computer system as recited in claim 15, wherein said second allocator allocates at least one object with a header that indicates said second promotion policy.

20. (Previously Presented) The computer system as recited in claim 15, wherein said header includes a preemption indicator indicating that a garbage collection count should be preempted and said at least one object should not be promoted to said next generation.

21. (Previously Presented) The computer system as recited in claim 20, wherein said header includes a preemption indicator and a preemption value;

said preemption indicator indicates that a garbage collection count should be preempted; and

said preemption value provides a preemptive garbage collection count that is used instead of said garbage collection count.

22. (Previously Presented) The computer system as recited in claim 20, wherein said header provides a garbage collection count that is used to determine when said at least one object should be promoted from Young Generation to said Older Generation.

23. (Previously Presented) The computer system as recited in claim 19, further comprising:

a garbage collector that reads said header and promotes said at least one object in accordance with said header.

24. (Previously Presented) The computer system as recited in claim 23, wherein said garbage collector delays or avoids promotion of said at least one object with respect to objects allocated with said first allocator.

25. (Previously Presented) The computer system as recited in claim 15, wherein said computer system is a virtual machine.

26. (Previously Presented) The computer system as recited in claim 15, wherein said computer system is a Java™ compliant virtual machine.

27. (Previously Presented) The computer system as recited in claim 15, wherein said computer system is provided for a handheld, an embedded, or mobile device.

28. (Cancelled)

29. (Previously Presented) A computer readable storage medium including computer program code for allocating objects in a memory portion that includes a Young Generation and at least one Older Generation, said computer readable medium including comprising:

computer program code for allocating one or more objects in said Young Generation in accordance with a first promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion; and

computer program code for allocating one or more other objects in said Young Generation in accordance with a second promotion policy exercised for postponing or cancelling promoting objects from said Young Generation to an Older Generation of said memory portion,

wherein,

each of said one or more objects includes a header that indicates said second

promotion policy.

30. (Previously Presented) The computer readable medium as recited in claim 29, wherein said second promotion policy is not to promote said one or more objects from said Young Generation to said Older Generation of said memory portion.

31. (Previously Presented) The computer readable medium as recited in claim 29, wherein said second promotion policy is to delay the promotion of said one or more other objects with respect to objects allocated under said first promotion policy.

32. (New) A computer readable storage medium including computer program code for allocating objects in a memory portion that includes a Young Generation and at least one Older Generation, said computer readable medium including comprising:

computer program code for allocating all of the objects into said Young Generation in accordance with either a first promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion or a second promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion, wherein the second promotion policy defines a longer tenure for the objects in the Young Generation than the first promotion policy; and

computer program code for promoting live ones of the objects allocated in the Young Generation to the Older Generation, the promoting being performed for each object involving first determining whether the object was allocated using the first or the second promotion policy and second applying the determined one of the first and second promotion policies.

33. (New) The computer readable medium as recited in claim 32, wherein said first promotion policy comprises assigning a value to a garbage collection count and wherein said second promotion policy comprises assigning a pre-emptive count value to the object that defines the longer tenure for the object in the Young Generation.